Beschreibung

Verfahren zur Überwachung des Programmlaufs in einem Mikrocomputer

Die Erfindung betrifft ein Verfahren zur Überwachung des Programmlaufs in einem Mikrocomputer eines elektronischen Gerätes, insbesondere einer Sensorschaltung für Kraftfahrzeuge, wobei das Programm Eingangsdaten verarbeitet und Ausgangsdaten erzeugt.

Bei elektronischen Geräten, die mit einem Mikrocomputer ausgerüstet sind, kann es durch Defekte im Mikrocomputer, insbesondere bei fehlerhaftem Programmlauf, zu Ausfällen und Fehlfunktionen kommen. Bei Sensorschaltungen für Kraftfahrzeuge und anderen sicherheitsrelevanten Geräten können derartige Fehler zu Gefährdungen führen, die mit sehr großer Sicherheit vermieden werden müssen.

Aufgabe der vorliegenden Erfindung ist es daher, den Programmlauf in einem Mikrocomputer möglichst vollständig und sicher zu überwachen und somit durch ein geeignetes Alarmsignal Benutzer zu informieren oder in verbundenen Systemen Gegenmaßnahmen einzuleiten.

Diese Aufgabe wird bei einer ersten Ausführungsform der Erfindung dadurch gelöst, dass zusätzlich zu dem Lauf des Programms eine Kopie des Programms, die in einem anderen Adressenbereich als das Programm im Mikrocomputer gespeichert ist, mit den für das Programm vorgesehenen Eingangsdaten abläuft und dass die Ausgangsdaten der Kopie mit denen des Programms verglichen werden und bei Nichtübereinstimmung eine Fehlermeldung erzeugt wird. Diese Ausführungsform berücksichtigt, dass bei einem an sich einwandfreien Programm ein Hardwarebedingter Fehler während des Ablaufs des Programms nicht auch bei einem identischen Programm auftritt.

Mit einer Weiterbildung kann die Kopie des Programms dadurch überprüft werden, dass ein weiterer Lauf der Kopie vorgesehen ist, bei welchem vorgegebene Testdaten verarbeitet werden, dass die Ausgangsdaten des weiteren Laufs der Kopie mit in einem Speicher abgelegten Vergleichsdaten verglichen werden und dass bei Nichtübereinstimmung eine Fehlermeldung erzeugt wird.

Um insgesamt den Lauf des Programms einschließlich des zur Überwachung dienenden Programms kontrollieren zu können, ist bei einer Weiterbildung des erfindungsgemäßen Verfahrens vorgesehen, dass nach dem Lauf des Programms und nach dem Lauf von Programmteilen, welche zur Durchführung des erfindungsgemäßen Verfahrens dienen, jeweils ein Flag gesetzt oder geändert wird und dass eine Fehlermeldung erzeugt wird, wenn nicht alle Flags gesetzt oder geändert wurden.

Eine zweite Ausführungsform der Erfindung dient zur Überwachung des Programmlaufs in mindestens zwei miteinander verbundenen Mikrocomputern eines elektronischen Gerätes, insbesondere einer Sensorschaltung für Kraftfahrzeuge, und besteht darin, dass in einem der Mikrocomputer eine Anfrage erzeugt wird, welche an den anderen Mikrocomputer gesendet wird und dort mit vorgegebenen Eingangsdaten den Lauf eines Programms bewirkt, dass eine von den Ausgangsdaten abhängige Antwort an den einen Mikrocomputer zurückgesendet wird und dass in dem einen Mikrocomputer die Anforderung und die Antwort miteinander verglichen werden. Auch hierbei kann wieder vorgesehen sein, dass das Programm eine Kopie eines die eigentliche Funktion des anderen Mikrocomputers ausführenden Programms ist.

Sowohl eine Überwachung desjenigen Computers, in dem das genannte Programm läuft, als auch eine Kontrolle der Überwachung ist bei der zweiten Ausführungsform der Erfindung dadurch möglich, dass die Antwort von Zeit zu Zeit verfälscht wird, was in dem anderen Mikrocomputer zunächst als Fehler

des einen Mikrocomputers erkannt wird, jedoch in dem einen Mikrocomputer erwartet und geprüft wird.

Auch hierbei kann der Programmlauf dadurch kontrolliert werden, dass nach dem Lauf des Programms und nach dem Lauf von Programmteilen, welche zur Durchführung des erfindungsgemäßen Verfahrens dienen, jeweils ein Flag gesetzt oder geändert wird und dass eine Fehlermeldung erzeugt wird, wenn nicht alle Flags gesetzt oder geändert wurden, wobei wiederum die Kontrollfunktion des anderen Computers dadurch überwacht wird, dass der Inhalt des Flagregisters von Zeit zu Zeit verfälscht wird, was in dem anderen Mikrocomputer zunächst als Fehler des einen Mikrocomputers erkannt wird, jedoch in dem einen Mikrocomputer erwartet und geprüft wird.

Da der jeweils andere Computer nicht zwischen Fehlern beim Programmlauf und den absichtlich von Zeit zu Zeit eingebrachten Fehlern unterscheiden kann, ist bei einer Weiterbildung vorgesehen, dass ein Fehlerzähler in einem der Mikrocomputer Fehler, die für den jeweils anderen Mikrocomputer festgestellt wurden, zählt und dass bei Einfügen einer falschen Antwort und/oder einer Verfälschung des Flagregisters der Zählerstand des Fehlerzählers in demjenigen Mikrocomputer, in dem die Einfügung der falschen Antwort bzw. die Verfälschung des Inhalts des Flagregisters stattfand, nicht geändert wird.

Die Erfindung lässt zahlreiche Ausführungsformen zu. Zwei davon sind schematisch in der Zeichnung anhand mehrerer Figuren dargestellt und nachfolgend beschrieben. Es zeigt:

Fig. 1: ein Blockschaltbild eines Anwendungsbeispiels für das erfindungsgemäße Verfahren,

Fig. 2: eine Darstellung verschiedener Funktionen zur Überwachung des Programmlaufs in einem Mikrocomputer,

Fig. 3: ein Flussdiagramm eines Programms zur Realisierung der in Fig. 2 erläuterten Funktionen und

Fig. 4: in Form eines Blockdiagramms die gegenseitige Überwachung zweier Mikrocomputer.

Im Zusammenhang mit den Ausführungsbeispielen wird das Programm auch Software-Routine genannt.

Das Anwendungsbeispiel gemäß Fig. 1 stellt einen Drehratensensor für ein Kraftfahrzeug dar, mit einem Vibrationskreisel 1, der Teil eines Sensormoduls 2 ist. Dies weist eine Reihe von Schaltungen zum Betrieb des Vibrationskreisels und zu der Auswertung der Signale des Vibrationskreisels auf, unter anderem auch einen Mikrocomputer 3. Dieser ist über einen SPI-Bus 4 mit einem weiteren Mikrocomputer 5 verbunden, der im folgenden auch Host genannt wird. Von diesem gelangt die Drehrateninformation über einen CAN-Bustreiber 6 an einen CAN-Bus 7 zur Weiterleitung an andere Systeme im Kraftfahrzeug.

Da es zum Verständnis der Erfindung nicht erforderlich ist, sind der Vibrationskreisel 1 und das Sensormodul 2 nicht näher erläutert. Wegen der Sicherheitsrelevanz des Drehratensensors ist eine Überwachung der ordnungsgemäßen Funktion der Mikrocomputer 3, 5, insbesondere des Programmlaufs, vorgesehen.

Bei dem in Fig. 2 dargestellten Beispiel sind diejenigen Funktionen, die der eigentlichen Funktion des Mikrocomputers (Primary function) dienen, als Rechtecke dargestellt. Bei 11 werden Eingangsdaten gelesen - beispielsweise aus den in Fig. 1 angedeuteten Schaltungen des Sensormoduls 2 - und in der zu überwachenden Software-Routine bei 12 verarbeitet. Die Ergebnisse dieses Programmlaufs werden bei 13 ausgegeben - im Falle von Fig. 1 letztendlich auf den CAN-Bus 7. Die betroffenen Mikrocomputer arbeiten in der Regel mit einer Reihe von Soft-

ware-Routinen, die sich zu einem Programmsystem ergänzen. In Fig. 2 ist die Überwachung einer Routine dargestellt, die bei Vorhandensein von mehreren Routinen besonders wichtig ist. Mit dem erfindungsgemäßen Verfahren können jedoch auch mehrere Routinen überwacht werden.

In dem Mikrocomputer 3, 5 (Fig. 1) ist außer der zu überwachenden Software-Routine in einem anderen Adressenbereich eine Kopie der Software-Routine abgelegt - im folgenden Kopie genannt. Zur Überprüfung des ordnungsgemäßen Programmlaufs der Original-Routine wird in einem ersten Schritt ein Lauf der Kopie mit den gleichen Eingangsdaten bei 14 durchgeführt. Die Ausgangsdaten dieses Programmlaufs werden mit den Ausgangsdaten der Original-Routine bei 15 verglichen. Im Falle von Unterschieden wird bei 16 ein Alarm ausgelöst.

In einem weiteren Schritt erfolgt ein Programmlauf der Kopie mit Testdaten 17. Die Ausgangsdaten dieses Programmlaufs werden mit gespeicherten erwarteten Ergebnissen, die in einer Look-up-table 18 abgelegt sind, bei 19 verglichen. Treten hier Unterschiede auf, wird ebenfalls bei 16 ein Alarm ausgelöst.

Zur Überwachung dessen, ob die in Fig. 2 dargestellte Überwachung auch tatsächlich stattfindet, ist vorgesehen, dass nach der Ausführung des Programmlaufs der Original-Routine bei 12 und nach Ausführung der Vergleiche bei 15, 19 Flags in einem Register 20 gesetzt werden. Bei 21 wird dann geprüft, ob alle Flags gesetzt sind. Ist dieses nicht der Fall, wird bei 22 ein Alarm ausgelöst.

Fig. 3 zeigt das bereits anhand von Fig. 2 erläuterte Überwachungsprogramm als Flussdiagramm, das beispielsweise alle 25 ms wiederholt wird. Dabei wird in einem ersten Programmschritt 31 zunächst die Original-Routine abgearbeitet, bei 32 folgt die Kopie, bei 33 werden die Ergebnisse verglichen. Im Programmschritt 34 wird dann die Kopie mit den Testdaten ab-

gearbeitet. Danach werden bei 35 die Ergebnisse des Programmlaufs 34 miteinander verglichen. Bei 36 wird geprüft, ob alle Flags gesetzt sind, worauf bei 37 das Flagregister initialisiert wird, d.h. rückgesetzt, wenn ein Setzen der Flags, wie im Zusammenhang mit Fig. 2 erläutert, durchgeführt wird. Alternativ kann anstelle des Setzens des Sets ein Toggeln durchgeführt werden.

Anhand von Fig. 4 wird die gegenseitige Überwachung zweier Mikrocomputer 3, 5 (Fig. 1) erläutert. Die langgestreckten Rechtecke 41, 42 stellen Datentelegramme auf dem SPI-Bus 4 (Fig. 1) dar mit jeweils einem Identifizierer, mehreren Nutz-Datenworten und einer Prüfsumme. Die ferner dargestellte Struktur ist auf beiden Mikrocomputern vorhanden, die sich gegenseitig überwachen.

Zur Überprüfung des jeweils anderen Mikrocomputers wird in dem einen bei 43 eine Aufforderung (Request index) generiert, die über den SPI-Bus 4 an den anderen Mikrocomputer übertragen wird. Dort werden aus einer Tabelle 44 Eingangsdaten für die zu überprüfende Software-Routine ausgelesen. Diese Daten werden in ein Programm 45 übergeben, das im Wesentlichen die Teile 11 bis 15 der Darstellung gemäß Fig. 2 enthält, d.h., in dieser Komponente führt der andere Mikrocomputer seine primäre Funktion aus und arbeitet ferner eine Kopie der Software-Routine ab.

Die Ausgangsdaten der Software-Routine werden in einer weiteren Tabelle 46 in eine Antwort (Response index) umgewandelt, die zu dem einen Mikrocomputer übertragen wird (siehe Datenwort 47 im Datentelegramm 42). Zweckmäßigerweise enthalten die Aufforderung und die Antwort nur jeweils einen Index, der besagt, welche in der Tabelle 44 abgelegten Eingangsdaten für die zu überprüfende Software-Routine verwendet werden sollen bzw. welchen Daten der Tabelle 46 die berechneten Ausgangsdaten entsprechen.

Der andere Mikrocomputer erhält eine Antwort (Datenwort 48 im Datentelegramm 41) und vergleicht diese bei 49 mit der erwarteten Antwort. Stimmen beide überein, wird angenommen, dass der andere Mikrocomputer in dieser Hinsicht richtig arbeitet. Treten jedoch Abweichungen auf, so wird ein Fehlerzähler 50 inkrementiert. Von Zeit zu Zeit werden in die Ausgangsdaten der Tabelle 46, also in die Antwort, Fehler eingefügt, die selbst bei richtigem Ablauf der Kopie 14 eine falsche Antwort 47 zur Folge haben.

Derjenige Mikrocomputer, der die falsche Antwort erhält, kann jedoch nicht erkennen, ob dies ein nachträglich eingeführter Fehler oder ein Fehler durch einen falsch ausgeführten Programmlauf ist. In ähnlicher Weise werden die im Zusammenhang mit Fig. 2 beschriebenen Flags bei 52 unterdrückt. Die Flags (in Fig. 4 als SR-Flags bezeichnet) werden bei 53 in das Datentelegramm 42 eingefügt. Sie führen bei dem anderen Programm vom Mikrocomputer im Programmteil 54 zu einer Fehlermeldung, wenn sie nicht alle gesetzt bzw. getoggelt sind.

Ferner wird der Zählerstand des Fehlerzählers 50 als weiteres Datenwort 55 in das Datentelegramm 42 eingefügt. Aus dem Datentelegramm 41 kann der Mikrocomputer das Datenwort 55 entnehmen und bei 56 überprüfen, ob der Zählerstnad dem erwarteten Wert entspricht. Trifft dies nicht zu, wird ebenfalls eine Fehlermeldung an den Fehlerzähler 50 gesendet. Die Funktion 56 erhält Meldungen von den Funktiomen 51, 52, wenn eine Verfälschung der jeweiligen Daten vorgenommen wurde, so dass dies bei dem Vergleich zwischen dem übertragenen Zählerstand und dem erwarteten Zählerstand berücksichtigt wird. Damit wird ermöglicht, dass durch die Funktionen 51 und 52 die Überwachung durch den anderen Mikrocomputer richtig durchgeführt wird. Der Zählerstand des Zählers 50 wird bei 57 daraufhin geprüft, ob ein vorgegebener Schwellwert erreicht ist. Ist dieses der Fall, wird bei 58 ein Alarm ausgelöst.

Patentansprüche

1. Verfahren zur Überwachung des Programmlaufs in einem Mikrocomputer eines elektronischen Gerätes, insbesondere einer Sensorschaltung für Kraftfahrzeuge, wobei das Programm Eingangsdaten verarbeitet und Ausgangsdaten erzeugt, dadurch gekennzeichnet, dass zusätzlich zu dem Lauf des Programms eine Kopie des Programms, die in einem anderen Adressenbereich als das Programm im Mikrocomputer gespeichert ist, mit den für das Programm vorgesehenen Eingangsdaten abläuft und dass die Ausgangsdaten der Kopie mit denen des Programms verglichen werden und bei Nichtübereinstimmung eine Fehlermeldung erzeugt wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass ein weiterer Lauf der Kopie vorgesehen ist, bei welchem vorgegebene Testdaten verarbeitet werden, dass die Ausgangsdaten des weiteren Laufs der Kopie mit in einem Speicher abgelegten Vergleichsdaten verglichen werden und dass bei Nichtübereinstimmung eine Fehlermeldung erzeugt wird.

3. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass nach dem Lauf des Programms und nach dem Lauf von Programmteilen, welche zur Durchführung des erfindungsgemäßen Verfahrens dienen, jeweils ein Flag gesetzt oder geändert wird und dass eine Fehlermeldung erzeugt wird, wenn nicht alle Flags gesetzt oder geändert wurden.

4. Verfahren zur Überwachung des Programmlaufs in mindestens zwei miteinander verbundenen Mikrocomputern eines elektronischen Gerätes, insbesondere einer Sensorschaltung für Kraftfahrzeuge, dadurch gekennzeichnet, dass in einem der Mikrocomputer eine Anfrage erzeugt wird, welche an den anderen Mikrocomputer gesendet wird und

dort mit vorgegebenen Eingangsdaten den Lauf eines Programms bewirkt, dass eine von den Ausgangsdaten abhängige Antwort an den einen Mikrocomputer zurückgesendet wird und dass in dem einen Mikrocomputer die Anforderung und die Antwort miteinander verglichen werden.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass das Programm eine Kopie eines die eigentliche Funktion des anderen Mikrocomputers ausführenden Programms ist.

6. Verfahren nach einem der Ansprüche 4 oder 5, dadurch gekennzeichnet, dass die Antwort von Zeit zu Zeit verfälscht wird, was in dem anderen Mikrocomputer zunächst als Fehler des einen Mikrocomputers erkannt wird, jedoch in dem einen Mikrocomputer erwartet und geprüft wird.

7. Verfahren nach einem der Ansprüche 4 bis 6, dadurch gekennzeichnet, dass nach dem Lauf des Programms und nach dem Lauf von Programmteilen, welche zur Durchführung des erfindungsgemäßen Verfahrens dienen, jeweils ein Flag gesetzt oder geändert wird und dass eine Fehlermeldung erzeugt wird, wenn nicht alle Flags gesetzt oder geändert wurden.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass der Inhalt des Flagregisters von Zeit zu Zeit verfälscht wird, was in dem anderen Mikrocomputer zunächst als Fehler des einen Mikrocomputers erkannt wird, jedoch in dem einen Mikrocomputer erwartet und geprüft wird.

9. Verfahren nach einem der Ansprüche 4 bis 8, dadurch gekennzeichnet, dass ein Fehlerzähler in einem der Mikrocomputer Fehler, die für den jeweils anderen Mikrocomputer festgestellt wurden, zählt und dass bei Einfügen einer falschen Antwort und/oder einer Verfälschung des

Flagregisters der Zählerstand des Fehlerzählers in demjenigen Mikrocomputer, in dem die Einfügung der falschen Antwort bzw. die Verfälschung des Inhalts des Flagregisters stattfand, nicht geändert wird.

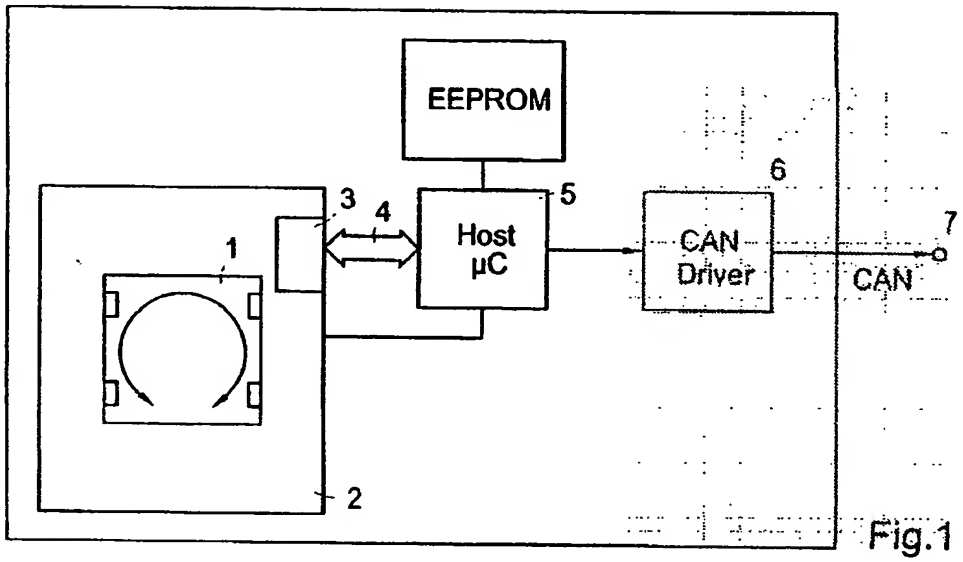
EEPROM
Host µC
CAN Driver
CAN
1
2
3
4
5
6
7

Fig.1

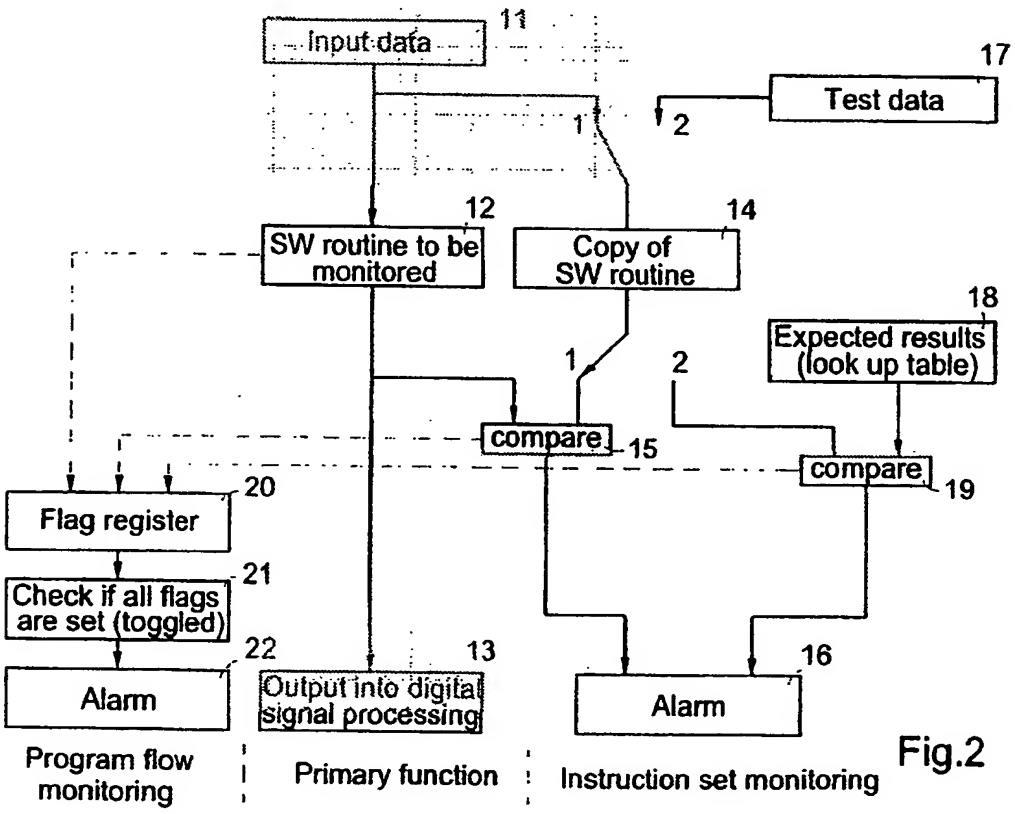
Input data
11
Test data
17
1
2
SW routine to be monitored
12
Copy of SW routine
14
Expected results (look up table)
18
compare
15
compare
19
Flag register
20
Check if all flags are set (toggled)
21
Alarm
22
Output into digital signal processing
13
Alarm
16
Program flow monitoring
Primary function
Instruction set monitoring

Fig.2

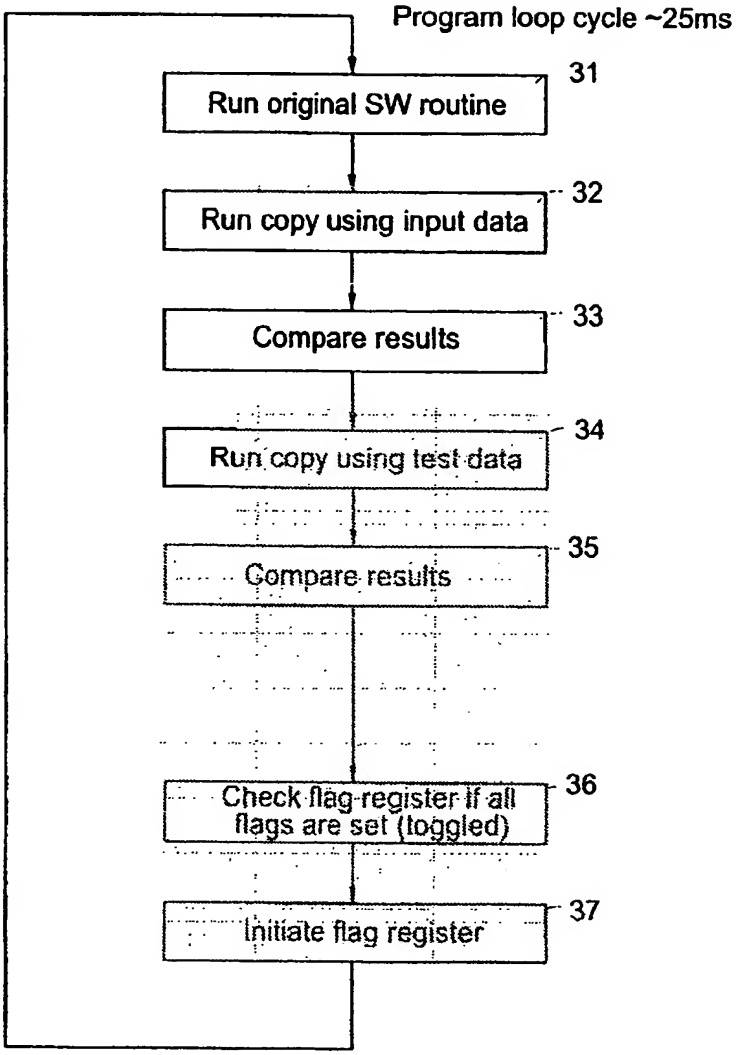
Program loop cycle ~25ms
Run original SW routine
31
Run copy using input data
32
Compare results
33
Run copy using test data
34
Compare results
35
Check flag register if all flags are set (toggled)
36
Initiate flag register
37

Fig.3

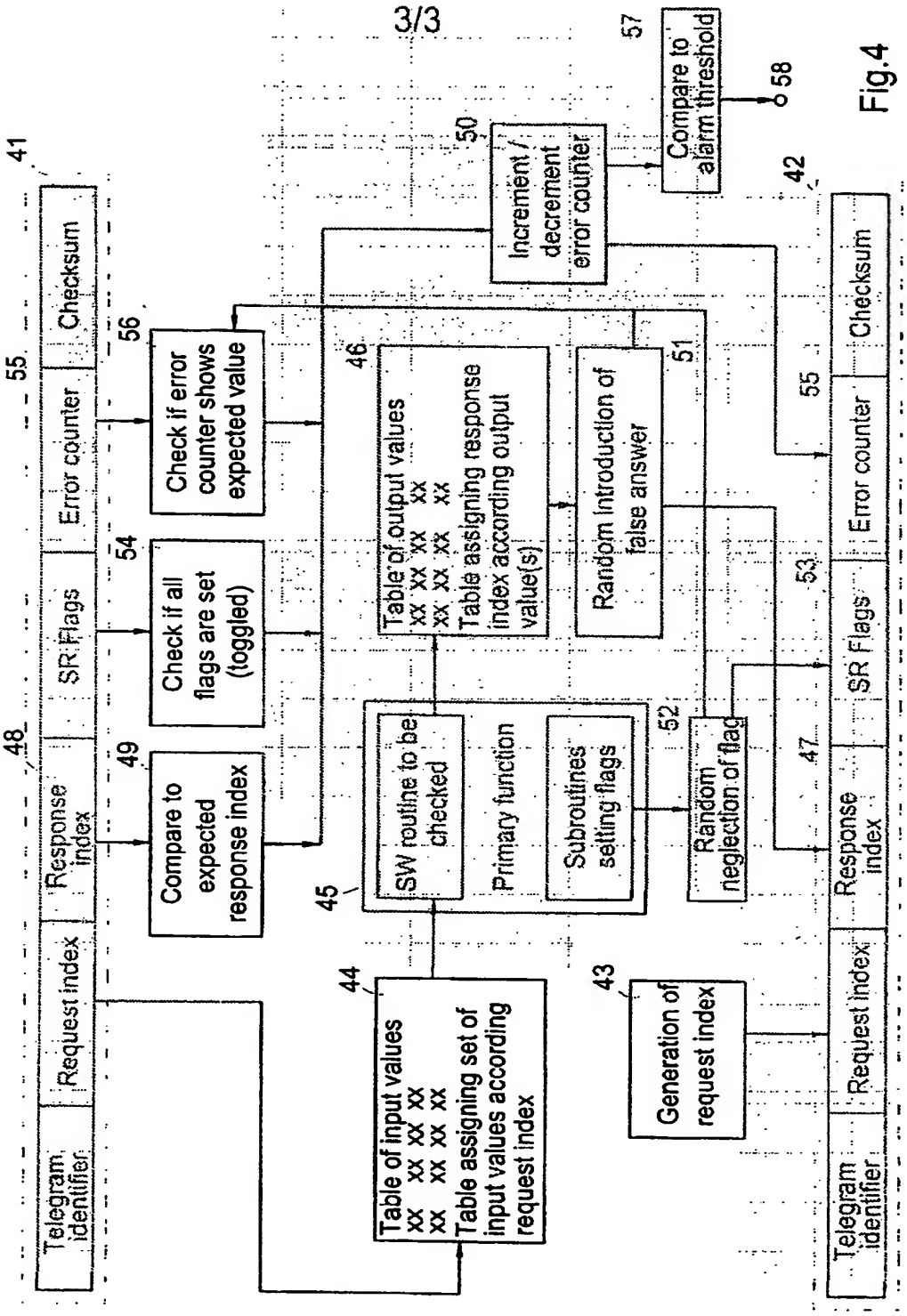
Telegram identifier
Request index
Response index
SR Flags
Error counter
Checksum
Telegram identifier
Request index
Response index
SR Flags
Error counter
Checksum
Table of input values
xx xx xx xx
xx xx xx xx
Table assigning set of input values according request index
Generation of request index
SW routine to be checked
Primary function
Subroutines setting flags
Random neglection of flag
Table of output values
xx xx xx xx
xx xx xx xx
Table assigning response index according output value(s)
Random introduction of false answer
Compare to expected response index
Check if all flags are set (toggled)
Check if error counter shows expected value
Increment / decrement error counter
Compare to alarm threshold
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

Fig.4